CLAIMS

What is claimed is:

1. An event monitoring component for dynamic optimization comprising:

an event monitor to selectively capture profiles of one or more microarchitecture events occurring in the execution of an application by a microprocessor based upon configuration information supplied by a software component; and

an interface through which the software component provides the configuration information to direct the operation of the event monitor.

2. The event monitoring component of claim 1, further comprising:

one or more monitor control vectors, the monitor control vectors storing the configuration information provided by the software component.

3. The event monitoring component of claim 2, wherein each monitor control vector includes:

a control field to specify a microarchitecture event to monitor;

a handler field, the handler field containing a pointer to a handler routine to process the profiles of the microarchitecture event; and

a trigger field to specify when the microarchitecture event is monitored.

4. The event monitoring component of claim 1, further comprising:

a profile buffer to store the captured profiles of the one or more microarchitecture events.

5. The event monitoring component of claim 4, wherein the profile buffer comprises a first level buffer for initial storage of the captured profiles of the one or more microarchitecture events and a second level buffer for subsequent storage of the captured profiles of the one or more microarchitecture events.

6. The event monitoring component of claim 5, wherein the first level buffer is a register file.

7. The event monitoring apparatus of claim 5, wherein the second level buffer is a memory buffer that is architecturally visible to the software component.

8. The event monitoring apparatus of claim 1, wherein the captured event profiles of each monitored microarchitecture events are made available to a handler routine selected by the software component for processing.

9. The event monitoring apparatus of claim 1, wherein the event monitoring apparatus initiates an interrupt or special event handler to notify the software component when captured event profiles are available for processing.

10. A microprocessor, comprising:

an execution pipeline;

one or more event monitors coupled to the execution pipeline to selectively monitor one or more microarchitecture events in the execution of a program and to capture event profiles;

one or more monitor control vectors to store configuration information provided by a software component in connection with the operation of the one or more event monitors; and

a profile buffer to store captured microarchitecture event profiles.

11. The microprocessor of claim 10, wherein the captured profiles of the one or more microarchitecture events stored in the buffer are made available to a handler routine selected by the software component for optimization processing.

12. The microprocessor of claim 11, wherein the profile buffer comprises a first level buffer for initial storage of the captured microarchitecture event profiles and a second level buffer for subsequent storage of the captured microarchitecture event profiles.

13. The microprocessor of claim 12, wherein the first level buffer is a register file.

14. The microprocessor of claim 13, wherein the second level buffer is a memory buffer that is architecturally visible to the software component.

15. The microprocessor of claim 14, wherein the first level buffer is comprised of a plurality of register frames and the second level buffer is comprised of a plurality of memory buffers, with one of the frames of the first level buffer and one of the memory buffers in the second level buffer being assigned to each monitored microarchitecture event.

16. The microprocessor of claim 15, wherein the profiles of a microarchitecture event stored in a frame assigned to the microarchitecture event in the first level buffer

memory are spilled into a buffer assigned to the microarchitecture event in the second level memory buffer when the frame assigned to the microarchitecture event in the first level memory buffer is fully allocated or when a condition established by the software component is met.

17. The microprocessor of claim 16, wherein the captured profiles of a microarchitecture event are made available to the handler routine when the buffer assigned to the event in the second level memory buffer is fully allocated or when a condition established by the software component is met.

18. The microprocessor of claim 10, wherein each of the monitor control vectors includes:

a control field to specify a microarchitecture event to monitor;

a handler field, the handler field containing a pointer to a handler routine for the microarchitecture event; and

a trigger field to specify when the microarchitecture event is monitored.

19. The microprocessor of claim 10, wherein the one or more microarchitecture events are monitored during an exception detection stage of the execution pipeline.

20. The microprocessor of claim 10, wherein the captured microarchitecture event profiles are stored in the memory buffer during a write back stage of the execution pipeline.

21. A method comprising:

receiving configuration information from a software component directing the monitoring of one or more microarchitecture events connected with the operation of a microprocessor in executing an application;

monitoring the one or more microarchitecture events and capturing profiles of the one or more microarchitecture events;

storing the captured event profiles in a profile buffer; and

making the profiles of the event available to the software component for optimization processing.

22. The method of claim 21, wherein receiving the configuration information from the software component comprises receiving information regarding the setting of one or more monitor control vectors.

23. The method of claim 22, wherein each monitor control vector includes at least the following fields:

a control field to specify the microarchitecture event to monitor;

a handler field, the handler field containing a pointer to a handler routine for processing of captured profiles of the microarchitecture event; and

a trigger field to specify when the microarchitecture event is monitored.

24. The method of claim 21, wherein the profile buffer is comprised of a first stage for initial storage of the captured profiles of the one or more microarchitecture events and a second stage for subsequent storage of the captured profiles of the one or more microarchitecture events.

25. The event monitoring component of claim 24, wherein the first stage is a register file.

26. The processor of claim 25, wherein the second stage of the profile buffer is a memory architecturally visible to the software component.

27. The method of claim 26, further comprising assigning a register in the first stage and a memory buffer in the second stage to each monitored microarchitecture event.

28. The method of claim 27, further comprising storing the profiles of each monitored microarchitecture event in the register assigned to the microarchitecture event in the first stage as the profiles of the microarchitecture event are captured.

29. The method of claim 27, further comprising spilling the profiles of a microarchitecture event from the register assigned to the microarchitecture event in the first stage to the memory buffer assigned to the microarchitecture event in the second stage when the register is fully allocated or when a condition established by the software component is met.

30. The method of claim 29, further comprising notifying the software component when the register assigned to the event in the second stage of the memory buffer is fully allocated or when a condition established by the software component is met.